Usability Testing:
Its Impact and Methodologies

Rasika Mohod
George Mason University
rmohod@gmu.edu

ABSTRACT

The success of a software product depends on its acceptance by the users. The launch of a product without considering its usability poses a significant risk to the business. Usability Testing tries to ensure that the product is usable by its users and that it succeeds. In this paper, I discuss various aspects of usability testing with a focus on why is usability testing important and how is it done. This paper will be useful for software testers and software engineers interested in usability testing.

Index Terms: Software Testing, Usability, Usability Testing, Heat Map

1 INTRODUCTION

In software engineering, usability is the degree to which a software can be used by specified customers to achieve specific goals with effectiveness, efficiency, and satisfaction in a specified context of use [13]. Usability has a huge impact on the software's success as user experience is an inherent design characteristic and requirement, which is closely related to various user errors [10]. The failure to take usability standards and principles into consideration leads to reduced user satisfaction and retention, thus affecting software revenue. For example, wrong placement of a form button on a major e-commerce site cost $300 million to the company [9]. It is, therefore, essential to create well-designed software that is highly usable. To evaluate if the software designed is usable, the designers can conduct usability testing. Usability testing is the way of assessing the product against customer reactions. Unlike software testing, the aim of usability testing is not to uncover code defects but to test if customers can actually use the product [1].

This paper discusses in detail different aspects of Usability Testing. This paper will be useful for software testing practitioners and software engineering researchers interested in usability testing. In this paper, I propose a detailed research on all dimensions of software usability testing and try to answer questions such as: What is Usability Testing? Why does it matter? When should it be conducted in software development life cycle? Who performs it? and How is it done?

I also discuss various online tools available in the market to perform usability testing. At the end, I propose the results obtained and its analysis through my experiments of usability tests performed on two software usability testing tools – TrymyUI [26] and UsabilityHub [27].

2 BACKGROUND

Contrary to other types of software testing which deal with functional behavior and code defects of the software, usability testing deals with user behavior and evaluates if the functionality can be used by the targeted users. It is thus wise to conduct usability testing before software testing, as any amendments to the functional specifications may be required to ensure that a desirable solution is built. And by the time the product reaches software testing, designers should be confident that the final product is usable [1]. Before I discuss in detail about usability testing, it is essential to first understand what is usability and usability testing. In this section, I give definitions of both the terms and briefly explain them with examples.

2.1 Usability

When new users first interact with a product, they do so with a specific goal in mind, that they are using the product because they want to achieve something and the product they have chosen offers a way to achieve it. *Usability* of the product describes the level of ease with which it allows a user to achieve that specific goal [3]. Usability consultant Jakob Nielsen and computer science professor Ben Shneiderman have written (separately) about a framework of system acceptability, where usability is composed of five criteria [4]:

1. Learnability: How easy is it for users to accomplish basic tasks the first time they encounter the design?
2. Efficiency: Once users have learned the design, how quickly can they perform tasks?
3. Memorability: When users return to the design after a period of not using it, how easily can they re-establish proficiency?
4. Errors: How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
5. Satisfaction: How pleasant is it to use the design?

For example, determining the usability of a university website means determining how easy is it for the users (e.g., students, professors, scholars, university staff, parents, academicians) to get desired information. If the product or in this case, the website has very confusing user interfaces, over-complex, multiple, and ambiguous task flows, poorly constructed error messages or bad performance, the result will be that the users will find it hard to achieve their goal [3]. This type of scenario describes poor usability. On the other hand, if the contrary happens and there is good and quick learnability, well-crafted user interfaces, straight-to-the-point task flows with good error messages and good performance, then the users will achieve their goal more easily and faster in time. Thus, this scenario describes good usability [3].

2.2 Usability Testing

Usability Testing refers to evaluating a product by testing it with representative users. The goal with usability testing is to recreate real world scenarios where the real users use the product [3]. By observing users' behavior product designers and developers identify potential usability problems, collect quantitative data on users' performance (e.g., time on task, task completion rate, error rates), and determine user satisfaction with the website [3].

For example, if a website is designed for a university, which allows users to retrieve all the information about university (e.g., academics, admissions, employment), the best way to test the design is by having users go through the website and developers observe users trying to access desired information. A usability test ensures that the interface of software under test (SUT) is built in a way that fits the user's expectations with respect to meeting requirements effectively, easily, efficiently, and in a satisfying manner [8].

3 WHY DOES USABILITY TESTING MATTER?

The more usable the product is, the more retention, customer satisfaction, and increased software revenue are achieved. Usability can thus be related to profit. The alternative to usability testing, which can be achieved by designers making guesses about user behavior, launching products without testing with real users, or not making changes in already launched products, results often in bad retention, poor satisfaction and consequently, less profit [3].

Let us consider the following usability failure example shared by Spool [9]. For a major e-commerce site, a login form was preventing customers from purchasing products to the tune of $300,000,000 a year. The login form was as simple as: two fields - Email Address and Password, two buttons - Login and Register, and one link - Forgot Password. The form was designed by the team to enable repeat customers to purchase faster and with an assumption that the first-time purchasers wouldn't mind the extra effort of registering. However, the problem was not with the form's design but was where it lived on the site. Users would encounter the form after they filled their shopping cart with products they wanted to purchase and pressed the Checkout button. The form came just before they could actually enter the information to pay for the product.

After the site was already launched, as a concern over less sales growth, the team decided to conduct a usability test. The test was conducted with real users who needed to buy products from the site. Users were asked to bring their own shopping lists and were given the money to make the purchases. All users needed to do was complete the purchase. However, the user behavior observed throughout the tests revealed user's complete different perspective for the form. First-time shoppers did mind registering. Some of the first-time shoppers couldn't remember if it was their first time, becoming frustrated as each common email and password combination failed. Many users even vocalized their views on registration, how the retailer only wanted their information to pester them with marketing messages and to invade privacy. Very few repeat customers remembered their login information and most stumbled on the form. When shoppers couldn't remember the email address and password, they'd attempt to guess, which rarely succeeded. Users would eventually use the Forgot Password link, which would again require them to remember which email address they initially registered with. An analysis of the retailer's database revealed some shocking results: 45% of all the customers had multiple registrations in the system, some as many as 10. Access to the link - Forgot Password - reached 160,000 per day and 75% of these people never tried to complete the purchase once requested.

However, the site asked nothing during registration that it didn't need to complete the purchase: name, shipping address, billing address, and payment information. The form, intended to make shopping easier, turned out to be helpful only for a small percentage of the customers. After analyzing the results, designers devised a simple fix to the problem. They took away the Register button. In its place, they put a Continue button with a simple message: "You do not need to create an account to make purchases on our site. Simply click Continue to proceed to checkout. To make your future purchases even faster, you can create an account during checkout." The number of sales observed after the fix came out went up by 45%. The extra purchases resulted in an extra $15 million the first month and for the first year, the site saw an additional $300 million.

This example shows how poor usability resulted in huge loss to company’s revenue. It proves that usability test helped designers to know the existence of the problem in their site that designers were unaware of. The usability test, in turn, resulted being one of the most important tests to detect the problem which could never be revealed by other software testing practices.

4 DIFFERENCES BETWEEN USER ACCEPTANCE TESTING AND USABILITY TESTING

When various types of software testing are talked about, the research has found that many designers do consider usability testing to be the same as user acceptance testing. Thus, designers fail to perform both testing practices which actually reveal different traits of the software. In this section, I discuss differences between user acceptance testing and usability testing.

User Acceptance Testing is used to find bugs in a website’s design and is performed in the last phase of Software Development Life Cycle (SDLC) [11]. It is generally used for the developers to prove to the client that the project is complete. Usability Testing, however, is used to test if an application is or will be easy to use for the end user. It should be done at the beginning, middle, and end of the design process, making incremental changes to the website design when issues are discovered [11].

For example, a team was developing a mobile game for Android—think Whack-A-Mole meets mutant veggies [11]. Eight months into the project they decided to do some user acceptance testing to find defects before launch. They got several reports of the game crashing from testers, but when looked upon into the code, the team couldn’t find out the reason. Thus, they decided to do some usability testing. After conducting usability testing with a designer, developer, note taker, recording equipment, and eight users it was determined that the crash wasn’t the issue. The spot where players had to drop veggies was dangerously close to the back button. Users kept hitting the back button by accident, causing the program to quit and make it seem as if the game had crashed. Here, as user acceptance testing dealt with functionality, it just showed that there was a critical error with the game, however, usability testing dealt with user behavior and revealed the true error. Each of the testing techniques is similar but have a very different goal. The biggest difference is that user acceptance testing verifies that the deliverable meets the agreed upon requirements whereas usability testing seeks to verify that an implementation's approach works best for the real user base.

5 WHEN IS USABILITY TESTING CONDUCTED?

The software testing literature recommends that the earlier a defect is found in the Software Development Life Cycle (SDLC) the cheaper it is to fix it [8]. The same concept holds true for usability testing. As discussed before, usability testing should be performed at the beginning, middle, and at the end of SDLC. The software undergoes many changes, interpretations, and implementations throughout the SDLC process. And to ensure not to make usability related mistakes at any of these steps, it is beneficial to conduct usability testing frequently and continuously throughout SDLC for maximum results [8]. On the other hand, when the relationship between functional and usability testing is considered, usability testing should be conducted again after functional testing, once the functional problems have been detected and fixed. This is recommended because performing usability testing on functionally-flawed products only unwinds functional problems. This hinders the usability test participants, thus negatively influencing their perception and invalidates the usability aspect of the results [12].

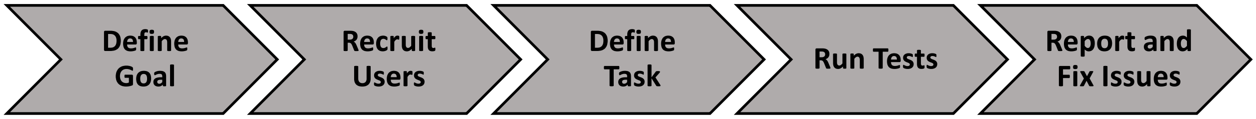
6 WHO PERFORMS USABILITY TESTING?

After understanding the importance of why usability testing is necessary and when it should be conducted, the next thing to focus on is who should perform usability testing that is who should be the testers, users, and participants of a usability test.

Unlike other software testing procedures, usability testing should not be performed by the designers and developers of the system. When designers and developers are working on a product, they become highly involved in that process, and that bias usually doesn't go away when they conduct software testing for the software themselves. It's borderline impossible for the same designer who designed a product to see all its flaws, after weeks and months thinking about it every day [8]. Since identifying those flaws is critical to software's success, the solution is to hire real time users and give them tasks. These real users should be hired in accordance with targeted audience criteria. A facilitator (designers or developers) who is conducting the test can devise these tasks and get the results from the users.

7 HOW IS USABILITY TESTING DONE?

Once all the pre-requisites of implementing usability test have been worked on, in this section, I discuss how do we perform the usability test. Its process involves five major steps: Defining Goal, Recruiting Users, Defining Tasks, Running Tests, and Reporting and Fixing Issues observed from test results [3]. This usability test process is shown in Figure 1.



Define Goal

Recruit Users

Define Task

Run Tests

Report and Fix Issues

Figure 1: Usability Testing Process

7.1 Defining Goal

The goal should be defined by clearly stating the need and motivation for a usability test. The goal definition process should consider the following scenarios [3]:

- Why is the test conducted and for which particular issues?
- What is the aim to be achieved?
- What are the success and failure criteria for the test?
- The hypothesis should be made about where most issues can be encountered?

7.2 Recruiting Users

The best way to test product's usability is to test it with real users that have had no part in the design or development process. However, recruiting those users is challenging, despite being one of the most crucial steps in the complete process of usability testing. Here, it is vitally important that users who are testing the product should fit the target audience. This fitness can be decided by taking various factors into account, such as geographical locations, age, gender, profession, education level, language. For example, if a product is built for a law firm for lawyers to use, it's not useful if junior engineers test the product [3]. In a usability test, the quality of users matters more than quantity in the recruiting process. Yet, the research on how many users to be used for a usability test has not been quantified. However, general assumption in the research area of usability testing is to use five users for one single usability test. As suggested by Neilsen et al. [6] [7], the best results come from testing no more than five users and running as many small tests that can be afforded. They showed that the number of usability problems found in a usability test with n users is: N(1-(1- L)^n), where N = total number of usability problems in the design and L = proportion of usability problems discovered while testing a single user. The typical value of L was 31%, averaged across a large number of projects studied by Nielsen et al. [7]. Plotting the curve for L =31% gave the following result as shown in Figure 2 [7].

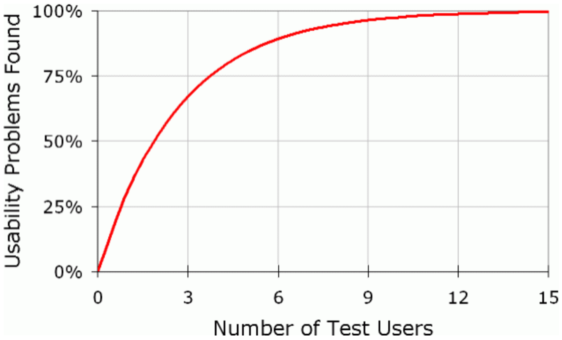


Figure 2: Curve plot by Neilsen

The results observed in this study showed that zero users give zero insights. A test with a single user showed that with first user insights shoot up and designers and developers learn almost a third of all there is to know about the usability of the design. The second user adds some amount of new insight, but not nearly as much as the first user. The third user will generate a small amount of new data, even if not as much as the first and the second user did. As more and more users are added, designers and developers learn less and less. After the fifth user, designers and developers are actually wasting their time by observing the same findings repeatedly but not learning much new. However, this theory of “five-user assumption” was contradicted by Faulkner [2]. The author suggested that five-user assumption theory is applicable and could reveal about 80% of all usability problems that exist in a product [6], only if all five users are not novices, but experts and will be best suited and fit all the requirements of targeted real users. According to the results of Faulkner’s study [2], each added users increased the minimum percentage of usability problems identified. Thus, comparing both these studies, it can be deduced that number of usability problems detected strongly depend on the team design. The number of users to be recruited to conduct usability test depend on budget, types of users available or accessible at that moment, the mission criticality of the system and the potential consequences of usability defects.

7.3 Defining Tasks

This step is another very important step in usability test process, as a task is the one which decides user’s behavior, results and thus further usability analysis. Defining tasks includes two steps:

1. Defining Task Scenario: A task scenario is the action that designers or developers ask the users to take on the interface to be tested [5]. Here, the aim is to provide users the context of the task to be achieved so users engage with the interface.
2. Defining Specific Task: This step aims at giving users actionable information required to perform the task.

For example, task scenario was “You're planning a vacation to Boston, June 3 – June 14. You need to buy both airfare and hotel.” And the task was “Go to the American Airlines site and Delta Airlines site and see who has the best deals.” [5] In this example, the task scenario provides user possible context of the test and the actual task describes the actions to be performed. Defining good task scenarios has a great effect on the usability test results. The poorly written tasks often focus too much on forcing users to interact with a specific feature, rather than seeing if and how the users choose to use the interface in general [5]. Such tasks kill the tests whereas well-written ones help to achieve the best results [3].

Given below are four task writing guides to improve the outcome of usability tests as suggested by Nielsen [5]:

7.3.1 Make the Task Realistic

Test participants should not be asked to do something that they wouldn’t normally do, this might make them try to complete the task without really engaging with the interface. Alternatively, participants can be allowed to define their own tasks. For example, recruit users who are in the process of buying a car and let users continue their research, instead of giving them a task scenario.

For example,	User goal:	Browse product offerings and purchase an item.
	Poor task:	Purchase a pair of green Reebok running shoes.
	Better task:	Buy a pair of shoes for under $50.

7.3.2 Make the Task Actionable

The users should be asked to do the action, rather than asking them how they would do it. The task can be termed as not actionable enough if the user turns to the facilitator, and seeks confirmation of their actions or asks for extra information.

For example,	User goal:	Find movie and show times.
	Poor task:	You want to see a movie Saturday night. Go to www.bookmyshow.com and tell me where you’d click next.
	Better task:	Use www.bookmyshow.com to find a movie you’d be interested in seeing on Saturday night.

7.3.3 Avoid Clues and Describing the Steps

The steps descriptions in the task should not contain hidden clues as to how to use the interface. Such tasks with clues bias users’ behavior and give less useful results.

For example,	User goal:	Look up grades.
	Poor task:	You want to see the results of your final exams. Go to the website, sign in, and tell me where you would click to get your transcript.
	Better task:	Look up the results of your final exams.

7.3.4 Avoid Vague Tasks

Avoiding clues should not make the tasks vague. If the task scenario is too vague, the users might ask for more information or might want to confirm if they are on the right path. Thus, users should be given all the needed information to complete a task, without telling them where to click.

For example,	Poor task:	Make an appointment with your doctor.
	Better task:	Make an appointment for next Thursday at 11am with your doctor, Dr. Peterson.

Here, the second task (better task) might seem violating the guideline for tasks to be realistic, if the user's dentist isn't really Dr. Petersen. However, this is one of those cases where users are very good at suspending disbelief and proceeding to make the appointment just as they would do with a differentlynamed doctor. The users might have to be asked to pretend to be seeing Dr. Petersen if usability testing is being performed on early prototype design, which only includes a few doctors.

7.4 Running Tests

The usability tests can be run either in a remote test environment with complete manual setup or by using usability testing tools. This section discusses the methodologies to be followed for running the tests in both the environments – Remote Testing and Using Usability Tools.

7.4.1 Remote Testing

Remote testing requires the tests to be conducted in presence of recruited users by giving them designed task scenario and specific tasks. The test results of such manual set up are obtained by observing the users throughout the test process and taking detailed notes of the user behavior. The user test sessions can also be visually recorded to preserve user's facial expressions as a resource for test output analysis. Users should be encouraged to use the "think aloud and write down" method to share their comments, feedback, and concerns. For example [1008], the feedback or comments can be requested by proposing questions: the task was successful or not, the task could be performed easily or not, and was the experience interesting, engaging or annoying.

7.4.2 Usability Testing Tools

There are various usability testing tools available in the market, which are beneficial to run usability tests, as these tools provide various methods to preserve test output and use it for further analysis. There is no clear criterion for categorization of these tools but it is mostly done on basis of the input given to tools and the test output provided by tool [8].

- Monitoring Tools
  These tools are used to monitor users while they perform the given task. Recruiting users and giving them tasks is a manual activity, outside of the tool. The idea is to create the tests and tasks and give them to users. While users are performing these tasks, the facilitator or developer watch their screen and interact with them [8].
  E.g., Skype [24], GotoMeeting [20], WebEx [31], TeamViewer [25]

- Task-Based Tools
  The tools of this category provide users, or designers and developers can pick users of their own choice by mentioning the requirements of the targeted audience that is real users. The product's page or design and the tasks to be performed must be submitted by designers and developers in the tool [8]. The tool, in turn, provides the videos of each user interaction along with the user's comments. This output can be used later to make usability analysis.
  E.g., OpenHallway [23], Whatusersdo [32], Userlytics [29], TrymyUI [26], UserTesting [30], Morae [22]

- Heatmap or User Movement Recorders
  These tools use eye tracking and heatmap methods to determine which part of the page the user has spent the most time on [8]. Some of the tools in this category also record the user's clicks, scrolls, mouse moves etc. Heatmap or mouse clicks of each user is delivered as output by the tool.
  E.g., UsabilityHub [27], CrazyEgg [16], FengGUI [17], Gazehawk [19], ClickHeat [14], ClickTale [15]

- Feedback or Survey Tools
  These tools provide with a feedback based on the website, page or wireframe that was submitted as input. Some tools of this type also provide surveys that help in giving conclusive evidence regarding usability issues [8].
  E.g., Kampyle [21], Usabilla [28], Five Second Test [18]

7.5 Reporting and Fixes

After all the tests are conducted, designers should revisit the interviewer's notes and re-watch each test several times. Designers should summarize the findings in a concise way, pointing to what were discovered to be the main pain points and note what were users' reactions and behavior. Successfully written test reports allow the team to work on the findings to improve the product. To fix the problems found, it's usually best to aim to fix the ones that require the least effort and deliver the most benefits. However, making the choice of which issues to be fixed and when, depends on the particular context and the specifics of the product [3].

8 EXPERIMENT

8.1 Test Settings

Two usability evaluations were conducted to study George Mason University's website (https://www2.gmu.edu/) using two different usability testing tools – TrymyUI [26] and UsabilityHub [27]. I chose these tools based on the type of test input accepted and the test output delivered by them. TrymyUI is a task-based tool, which delivers output in the form of a video for each user interaction. UsabilityHub is a heatmap and user movements recorder and delivers output in various forms such as heatmap, mouse clicks, feedback questions, demographics that include user's country, age, and education level. Unlike TrymyUI which provides only one type of task-based test, UsabilityHub provides five types of tests to choose from: Question Tests: To gather quick feedback on designs, Five Second Tests: To measure messaging clarity within five seconds, Click Tests: To visualize how users interact with designs, Navigation Tests: To analyze how users navigate through websites, and Preference Tests: To verify which design resonates by collecting preferences from users. The first usability evaluation was conducted using the task-based test in TrymyUI and the second evaluation was conducted using navigation test in UsabilityHub. Both the tools are easy to use online paid tools with necessary tutorials provided by tools.

The goal set for the usability test was, "to access information about different academic programs offered by George Mason University (GMU)." The motivation behind setting this goal was to verify if the GMU website is usable for students to access required academic information. Two different tasks sets were designed to perform usability evaluation on two different tools. The aim of the tasks was to verify if users could retrieve the web page of given specific academic program "Masters of Science in Software Engineering" of GMU. The task completion required five clicks on the website to successfully retrieve "Masters of Science in Software Engineering" web page. The success criterion was set as following navigation pattern: GMU Home Page -> Click 1: Academics Tab -> Click 2: Graduate Studies Menu -> Click 3: Software Engineering, MS Option -> Click 4: CS Department Home Page -> Click 5: MS in Software Engineering Menu. The failure criterion was set as user performing at least one unsuccessful click.

8.2 Test Users

In both usability evaluations, the users provided by tools were used to perform usability tests. No prior requirements were mentioned in the tool to pick specific users.

8.3 Usability Test 1: TrymyUI

The first test was conducted with TrymyUI tool. The test was conducted with one user provided by the tool. The tool provided a task-based test, which required test scenario and tasks as input for the test. The test output was delivered by the tool in the form of video recorded of user's interaction with the website.

8.3.1 Test 1 Input

Task Scenario: "You are an aspiring Graduate student and want to know more about graduate courses offered by George Mason University, VA."
Task: "Search for a graduate course "MS in Software Engineering."

8.3.2 Test 1 Results

The video provided by tool showed that the user failed to execute the given task successfully. The user's first click was successful, however, after first navigation user completely deviated from the expected navigation path. Thus, success rate of this test was calculated as 0%. Table 1 shows the detailed test results with successful responses (successful clicks) for each of the five clicks. The successful responses are specified as actual count and percentage.

Table 1: TrymyUI Test Results

	Click 1	Click 2	Click 3	Click 4	Click 5
Successful Responses	1 (100%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)

8.3.3 Test 2 Analysis

After observing the video, the user's comments and thinking out loud methodology helped to understand the user's behavior. The user was focused on searching courses offered by GMU instead of searching for the graduate program "Masters of Science in Software Engineering". This confusion was created as the task scenario mentioned, "…graduate *courses* offered by George Mason University," which should have been written as "… graduate *programs* offered by George Mason University". Another point where user stumbled upon in video was while interpreting the acronym "*MS*" described in the task. The personal information about the user provided by the tool revealed that she was a high school graduate residing in the remote area of US. Thus, her education background could be the reason for not interpreting the acronym "MS". This analysis revealed another loophole in the task description that is the use of acronyms, which made the task unclear.

The analysis performed on the test results justified the failure of usability evaluation. This motivated me to conduct second usability evaluation with modified test scenario and task description.

8.4 Usability Test 2: UsabilityHub

The second test was conducted using navigation test with UsabilityHub tool. The test was conducted with 25 users provided by the tool. The navigation test required images or screenshots of each of the five web pages involved in successful test execution to be provided as test input with the hitzones mentioned. The hitzone for UsabilityHub tool is the selected area on the image or screenshot, which represents successful clicks. The tool provided a detailed summary of test results in the output, which included: average duration of test; success rate, failure rate, and average time taken for each click (on each image); heatmap and user clicks on each image; users' comments on unsuccessful click (asking why did they click there); demographics such as number of users from different countries, age groups, gender, and education level.

8.4.1 Test 2 Input

Task Scenario: You are an aspiring graduate student and want to learn more about graduate programs offered by George Mason University, Virginia, US.
Task: How would you search the graduate program "Master of Science (MS) in Software Engineering" on George Mason University website?

8.4.2 Test 2 Results

The detailed test results report showed that out of 25 users only 3 users could successfully execute the task, thus success rate of 12% was observed. The average duration of test execution was 72.6 seconds. Table 2 given below shows the detailed test results with successful responses (successful clicks) for each of the five clicks. The successful responses are specified as actual count and percentage.

Table 2: UsabilityHub Test Results

	Click 1	**Click 2**	**Click 3**	**Click 4**	**Click 5**
Successful Responses	16 (64%)	7 (28%)	3 (12%)	3 (12%)	3 (12%)

8.4.3 Test 2 Analysis

The heatmap and user's clicks provided for each image in the test results provided a concise view of what made 88% of users to perform unsuccessful task execution and navigate to undesired paths. On the first GMU home page image, out of 36% unsuccessful clicks, 16% clicks were made on "Admissions & Aid" tab, 4% on "Prospective Students" menu, 4% on "Research" tab and remaining on other points on the web page. Similarly, out of 56% (9 out of 16) of, 64% users who succeeded in the first click, made unsuccessful clicks on "Major and Minors", "Student Life", and "Find Your Area of Study". Further 12% of users who succeeded the first two clicks completed the task execution successfully.

It was thus observed that first two web pages encountered by users possessed various similar meaning labels and names given to menus, tabs, and other options listed on the page. The unsuccessful clicks on the first page such as "Admissions & Aid", "Prospective Students", and "Research" were similar in meaning to the expected tab selection "Academics". Also, unsuccessful clicks from the second web page which included "Major and Minors", "Student Life", and "Find Your Area of Study" were found to be similar and confusing with the expected menu selection that is "Graduate Studies". This analysis thus deduced that usability of GMU website was hindered mainly due to confusing labels and names of tabs and menus and presence of multiple task flows.

9 CONCLUSION AND FUTURE RESEARCH

Software usability plays an important role in shaping software revenue, thus usability test must be performed by all software products throughout its development life cycle. Though usability test varies by organization and product, the general purpose of every usability test should be to test the product before launch by mimicking a real world as much as possible. To perform usability test, recruit representative users and ensure that task scenario defines the context of the test and each task is realistic, actionable, and clear without giving any clues and step descriptions. The end results or outcome of a usability test are not the actual defects. The usability test output is the information on how users interacted with the interface, what they liked, what they did not and is basically raw data. Usability tester's job thus does not end with reporting the results but includes analyzing the results, reporting the usability issues in the interface and fixing them. There are various online usability testing tools available, which should be used to perform testing with ease. These tools provide users or participants from worldwide thus providing a real-world scenario for the product.

The experiment conducted using two usability testing tools, TrymyUI and UsabilityHub showed the effects of appropriately designed test scenario and task definitions on the success rate of a test. The poorly designed tasks can provide false test results thus invalidating the aspect of usability testing. The properly designed test inputs provided to the tools revealed important and unknown usability issues of the tested website.

So far, usability testing is implemented in the remote test setup and using different testing tools. There are various types of tests which can be undertaken depending on the type of usability issues to be detected or the type of results obtained through tests. The research area to map appropriate methodology and type of usability tests to be conducted with each type of possible usability issue can be studied further. The usability testing methodology can thus be formalized to help software testing practitioners.

REFERENCES

[1] Coster, Nick, "Usability Testing vs Software Testing," in *Brainmates, Product Management Training and Expertise Framework* https://brainmates.com.au/brainrants/usability-testing-vs-software-testing/

[2] Faulkner, Laura, "Beyond the five-user assumption: Benefits of increased sample sizes in usability testing," Published on May 18, 2003

[3] Muniz, Fabio, "What is Usability Testing?," in *Testlodg: Online Test Case Management Tool*, Published on Sep 14, 2016 http://www.softwaretestinghelp.com/usability-testing-guide/

[4] Nielsen, Jakob, "Usability 101: Introduction to Usability," in *Jakob Nielsen's Alertbox, Nielsen Norman Group,* Published on Jan 4, 2012 https://www.nngroup.com/articles/usability-101-introduction-to-usability/

[5] Nielsen, Jakob, "Turn User Goals into Task Scenarios for Usability Testing," in *Nielsen Norman Group,* Published on Jan 12, 2014 https://www.nngroup.com/articles/task-scenarios-usability-testing/

[6] Nielsen, Jakob and Landauer, Thomas K., “A mathematical model of the finding of usability problems,” in Proceedings of ACM INTERCHI '93 Conference (Amsterdam, The Netherlands, 24-29 Apr 1993), pp. 206-213.
https://dl.acm.org/citation.cfm?id=169166

[7] Nielsen, Jakob, “Why You Only Need to Test with 5 Users,” in *Nielsen Norman Group,* Published on Mar 19, 2014
https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/

[8] Seela, Swati, “A Complete Guide to Usability Testing – It’s Like Trying to Read Minds!,” in *Software Testing Help, GUI Testing,* Published on Apr 17, 2017
http://www.softwaretestinghelp.com/usability-testing-guide/

[9] Spool, Jared M. “The $300 Million Button,” in *User Interface Engineering (UIE),* Published on Jan 14, 2009
https://articles.uie.com/three_hund_million_button/

[10] Tana, Wei-siong and Liub, Dahai and Bishua, Ram, “Web evaluation: Heuristic evaluation vs. user testing,” in *International Journal of Industrial Ergonomics*, Volume 39, Issue 4, Jul 2009, Pages 621–627
http://www.sciencedirect.com/science/article/pii/S016981410800053X

[11] Virothaisakun, Joël, “User Acceptance Testing Versus Usability Testing,” in *DigitalGov,* Published on Oct 6, 2014
https://www.digitalgov.gov/2014/10/06/user-acceptance-testing-versus-usability-testing-whats-the-dif/

[12] “Functional Testing vs. Usability Testing,” in *QualiTest*
https://www.qualitestgroup.com/white-papers/functional-testing-vs-usability-testing/

[13] “Usability,” in Wikipedia
https://en.wikipedia.org/wiki/Usability

Usability Testing Tools:

[14]	ClickHeat	http://www.clickheat.eu/
[15]	ClickTale	https://www.clicktale.com/
[16]	CrazyEgg	https://www.crazyegg.com/
[17]	FengGUI	https://www.crazyegg.com/
[18]	Five Second Test	http://fivesecondtest.com/
[19]	Gazehawk	http://gazehawk.com//
[20]	GoToMeeting	https://www.gotomeeting.com/
[21]	Kampyle	http://www.kampyle.com/

[22]	Morae	https://www.techsmith.com/morae.html
[23]	OpenHallway	http://www.openhallway.com/
[24]	Skype	https://www.skype.com/en/
[25]	TeamViewer	https://www.teamviewer.com/en/
[26]	TrymyUI	https://www.trymyui.com/
[27]	UsabilityHub	https://usabilityhub.com/
[28]	Usabilla	https://usabilla.com/
[29]	Userlytics	https://www.userlytics.com/sitepublic/
[30]	UserTesting	https://www.usertesting.com/
[31]	WebEx	https://www.webex.com/
[32]	Whatusersdo	https://www.whatusersdo.com/